# Autonomous System Technologies for Resilient Airspace Operations

Vincent E. Houston[1] and Lisa R. Le Vie[2]

NASA Langley Research Center, Hampton, VA, 23666

**Increasing autonomous systems within the aircraft cockpit begins with an effort to understand what autonomy is and developing the technology that encompasses it. Autonomy allows an agent, human or machine, to act independently within a circumscribed set of goals; delegating responsibility to the agent(s) to achieve overall system objective(s). Increasingly Autonomous Systems (IAS) are the highly sophisticated progression of current automated systems toward full autonomy. Working in concert with humans, these types of technologies are expected to improve the safety, reliability, costs, and operational efficiency of aviation. IAS implementation is imminent, which makes the development and the proper performance of such technologies, with respect to cockpit operation efficiency, the management of air traffic and data communication information, vital. A prototype IAS agent that attempts to optimize the identification and distribution of "relevant" air traffic data to be utilized by human crews during complex airspace operations has been developed.**

## Nomenclature

| | |
|---|---|
| *AC* | = Aircraft |
| *ATC* | = Air Traffic Control |
| *Bagging* | = Bootstrap Aggregating |
| *IAS* | = Increasingly Autonomous Systems |
| *FDIT* | = Flight Deck Interface Team |
| *LC* | = Light Craft |
| *ML* | = Machine Learning |
| *NAS* | = National Airspace System |
| *ND* | = Navigation Display |
| *NextGen* | = Next Generation Air Transportation System |
| *PM* | = Prediction Model |
| *SASO* | = Safe Autonomous Systems Operations |
| *SSL* | = Sound Source Localization |
| *TDM* | = Traffic Data Manager |
| *TDM-PM* | = Traffic Data Manager - Prediction Model |
| *VISTAS* | = Visual Imaging Simulator for Transport Aircraft Systems |

[1] Computer Research Engineer, Crew Systems Aviation and Operations, NASA Langley, Senior Member
[2] Aerospace Research Engineer, Crew Systems Aviation and Operations, NASA Langley, Member

## I. Introduction

Due to ever-increasing operational rates within the National Airspace System (NAS) there is a need to explore research and develop technologies that implement the Next Generation Air Transportation System (NextGen) Autonomic Airspace Architectures.[1] One of the Safe Autonomous Systems Operations (SASO) Project's objectives focusses on the development of system technologies that can overcome current-day human performance limits and enable applications, functionality, and performance that are not possible today. These applications include improving a flight crew's awareness of traffic and intent, while maintaining or decreasing crew workload, even as traffic increases. In addition, SASO endeavors to facilitate the operational transformation of applications that would showcase how Increasingly Autonomous Systems (IAS) could prove beneficial to the enhancement of the safety and efficiency of civil aviation. [1]

Increasing autonomous systems within the aircraft (AC) cockpit begins with an effort to understand what autonomy is and developing the technology that encompasses it. Autonomy allows an agent, human or machine, to act independently within a delineated set of criterion; delegating responsibility to the agent(s) for the achievement of the overall system objective(s). IAS are the maturation of current automated systems (i.e., the relegation of tasks) toward full autonomy (i.e., the delegation of responsibilities). Working in concert with humans, these types of technologies are expected to improve the safety, reliability, costs and operational efficiency of aviation, and must do so without resulting in any degradation to current system safety performance. Achieving these expectations, with acceptable accuracy, builds a rapport between human and machine. This is important; because, in order for humans and autonomous systems to dynamically adjust and cooperate with each other to accomplish a joint objective (Human-Autonomy Teaming) the human has to trust the machine's abilities. A flight crew's trust can be attained when the autonomous system's predictions are highly accurate and that accuracy is consistent.

An IAS agent prototype has been developed that identifies "relevant" air traffic data to be utilized by humans during autonomous, or increasingly autonomous computer-to-computer interactions during complex connected airspace operations. The Traffic Data Manager (TDM) application -a SASO-derived IAS technology- is designed to prevent a flight crew from being overwhelmed by the sheer volume of varying forms of information, inherent to a Net-Centric or connected airspace environment by creating an "intelligent" party-line. Within the cockpit, TDM should prove invaluable to the flight crew by providing only pertinent information (traffic, intent, messaging, and state/status data) that would ensure a crew's oversight, awareness, and intervention/interaction potential remain intact. The TDM employs machine learning to meet and augment the informational needs of a flight crew, based upon expert human judgement, while adapting to changing environments and operations. The expectation is that the TDM application, utilized in the area of air traffic and data communication information management, can effectively declutter a flight crew's visual and auditory sense workload; thereby, creating a more efficient and effective cockpit environment.

## II. Study Design

This paper describes the TDM's prediction model (TDM-PM) and an evaluation of its accuracy. Also described is how the model's predictions are used to improve cockpit situation awareness. This foundational work was critical to the development of TDM, specifically, and IAS, in general, in terms of assessing how to capture a pilot's expert knowledge; in this case, how to determine air traffic relevance and to train a machine learning (ML) algorithm to make like inferences on relevance based on aircraft state and contextual data. This work was a multi-system technological effort. The following components and parameters were identified and developed into one application.

### A. Machine Learning Algorithm

Primary to the development of TDM was the selection of the best algorithm capable of learning without definitively programming it or over-constraining it toward a specific result. In the case of TDM, that result is the determination of relevant air traffic with respect to ownship.

There were several types of ML algorithms that could have been used for this technological development: Supervised, Semi-Supervised, and Unsupervised. Semi-Supervised and Unsupervised algorithms had benefits with respect to incomplete or nonexistent training data, leaving the algorithm itself to establish structure and conclude if any hidden patterns exist. However, Supervised learning, as a parsing method, proved to be the most appropriate, because categories are a natural consequence of the application (i.e., traffic is either *relevant, maybe-relevant, and not-relevant* to the pilot/operation).

The supervised process of learning from a training dataset is akin to an instructor supervising the learning process of a student. In the case of TDM's ML algorithm, the instructor is the expert pilot community and the student is the algorithm itself. Using the Dynamic Air Traffic Application (D.A.T.A) [3] to display aircraft state and traffic scenarios,

pilot observations/responses to the relevancy of the traffic were collected, forming a ≈22K knowledge base used for Supervised ML training and model prediction testing. The algorithm creates a model that allows predictions to be made concerning these air traffic relevancy data with its associated state data. As new observations are introduced, the algorithm's ability to predict progressively improves.[2] The formulation of the TDM prediction model is illustrated in Figure 1.

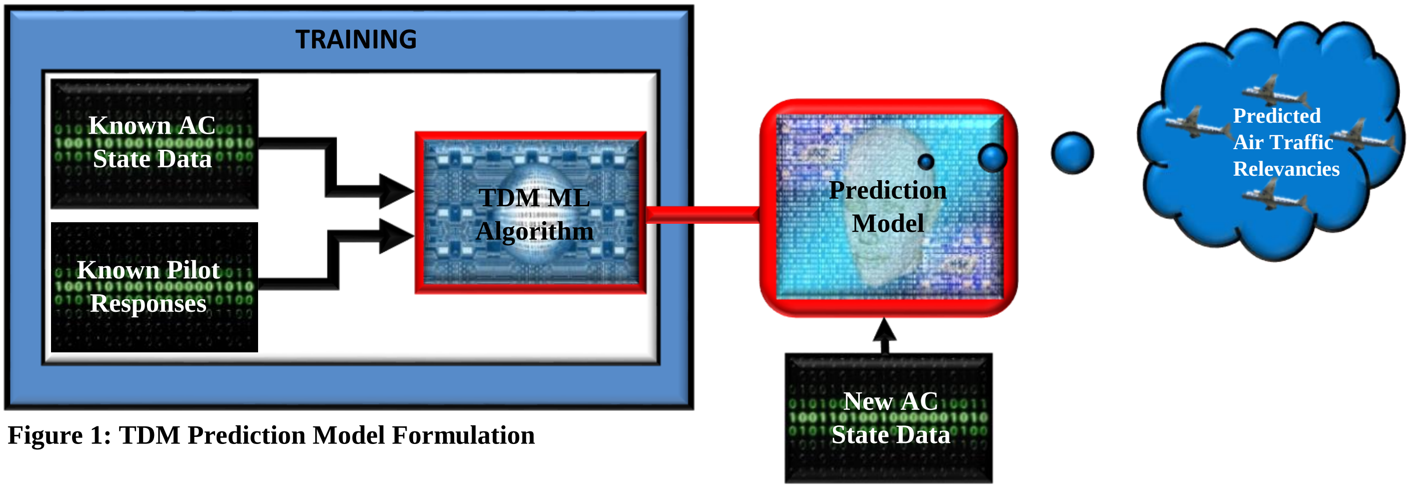


**Figure 1: TDM Prediction Model Formulation**

### B. Supervised Data – Data Collection Design

Essential to the application of the ML algorithm was the collection of data that would be used to train the algorithm. The Dynamic Air Traffic Application (D.A.T.A) was developed to integrate air traffic state data information into the Electronic Flight Bag (EFB) /tablet computer framework for data collection using the following data labels (predictors) *traffic course, airspeed, altitude, rate, range, and bearing* with respect to ownship (see Figure 2 for example).

These data were displayed to simulate an air traffic environment with respect to ownship onto D.A.T.A's Navigation Display (ND) format, allowing Subject Matter Experts (SME) to quickly evaluate various traffic configurations and assign relevance (i.e., assign either relevant, not relevant, or maybe relevant labels) to each traffic element.

Each training scenario was executed with ownship being located differently in the airspace, with the objective of trying to cover as much of a 360° circumference as was thought necessary to train the algorithm. The AC state data became the data labels that were used to train the TDM-ML algorithm, along with a relevancy marker input by the SMEs specifying each aircraft's importance with respect to ownship. Details of this study's data collection are contained in reference 3.

### C. Classification TreeBagger Algorithm

The overall objective of TDM's ML algorithm was to learn how to perform a task that today is strictly reserved for implementation by the human agent; that is, the assessment of air traffic with respect to ownship. The ML algorithm best suited for the development of TDM was the Classification instance of ML. Several types of the classification algorithms were tested (K-Nearest Neighbor, Naive Bayes, and Support Vector Machine). However, the ensemble classifier method of blending results from weak learners into very high-quality predictors proved to be the most consistently robust and accurate of the algorithms tested.

Utilizing the Bootstrap-aggregated (Bagged) method of ensemble learning 75% of the input data set (AC state data and pilot responses) were used for random uniform distribution, through draw with replacement, to teach the algorithm. The algorithm repeated the draw-and-replace process several times, which resulted in the creation of several classifiers. Voting, or polling, between the created classifiers produced the final prediction of air traffic relevance. The TreeBagger[3], TDM's ML algorithm, integrated decision tree learning with the bagged ensemble classifier method to provide a very robust prediction model that reduced the effects of over-fitting, inherent to decision trees, and improved the model's ability to accurately and consistently predict air traffic relevance based on inferences from the data set.[4,5]

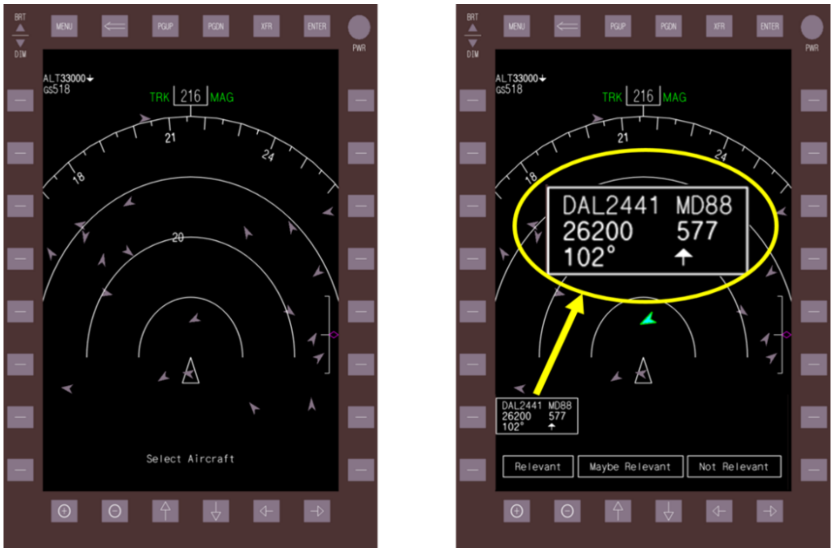


**Figure 2. Dynamic Air Traffic Application (D.A.T.A)**

### D. Study Objectives and Variables

Currently, there are no aviation standards in place to address the concept of trust between machine and human within the aviation domain. This study, as a one-to-one comparison between TDM-PM relevancy forecast and the correct relevancy determinations made by human pilots, makes it possible to use the results to lay the groundwork for identifying what could prove to be trust factors. The trust or confidence in TDM-PM should be proportional to the commonality of its responses with those of the SMEs.[11,12]

With that in mind, an accuracy rate for the TDM-PM responses compared to the SMEs responses was established. At this stage of development (prototyping), the TreeBagger prediction accuracy was set to 90%. MATLAB's TreeBagger prediction function produces "scores" generated by each tree that represents the probability of this observation originating from a particular class and averages the scores over all trees in the ensemble.[2,3] It was concluded that, if the TDM-PM accuracy rate could be achieved, then not many enhancements to the training process or additional data labels/predictors would be needed in order to solidify the machine-human collaboration and trust.

Another potential trust factor of interest is *model prediction latency*, which is the time it takes to render a prediction given a query. Because predicting is a critical path element, predictions have to be fast and require bounded tail latencies. Sophisticated models, like those produced from the TreeBagger algorithm, are computationally complex, and can therefore be substantially latent[7,8]. Because the closing rate of aircraft can require resolutions in seconds, it is a necessity that a prediction model be able to generate predictions expeditiously. We benchmarked that the TDM-PM produces an inference based on the AC state data input in no more than three seconds.

## III. Results and Discussion

The focus of this study was to statistically compare the responses of the human-pilot with those of the TDM-PM and explore how to capture a pilot's expert knowledge and train a ML algorithm to make similar inferences as a pilot based on aircraft state and contextual data.

For the TDM-PM to be deemed valid, we assumed a 10% differential between its responses and those of the human-pilot to be expected. This expectancy was based on the three-to-one ratio between the training and testing data, respectively, that we used for the TDM-PM ML algorithm development. Other ratios were used 80/20 and 90/10. However, the 75/25 showed the best performance. A total of approximately 22K data points formed the baseline/control data for this effort.

The results showed that the TDM-PM resulted in the model's overall error-accuracy rate of 74%. This may be acceptable for some applications of the algorithm; however, a more robust determination of the model's accuracy, and one that would indicate the TDM-PM as being a useable representation of the human-pilot is needed.

### A. System Model Response Comparison

The TDM-PM was developed to address a future increase of autonomous system technology within the cockpit of commercial aircraft operating in a NextGen NAS. The most substantive way of evaluating TDM-PM's soundness (validity) would be to verify that its output responses directly to the responses of the SMEs (human-pilot system).

Figure 3 provides an overall look at the system-model relevancy comparison from the ≈22K baseline data points. The data shows that TDM-PM predicted 19,640 non-relevant AC opposed to 17,463 non-relevant determinations provided by the human-pilot system. This resulted in TDM-PM correctly matching 88.7% of the systems non-relevant AC determinations and, at the same time categorizing, 12% more non-relevant AC than was determined by the human-pilot.

With respect to the *maybe-relevant* data points, the TDM-PM predicted 1756 of the human-systems 3258 *maybe-relevant* objects of air traffic. Within TDM-PM's 1756 *maybe-relevant* predictions, 1581 of them matched the determinations made by the human-pilot system.

Finally, TDM-PM's *relevant* predictions was 54% of the determinations provided by the SMEs. However, within the 790 relevant predictions, the PM matched the human-pilot system 86.1% of the time.

These data demonstrated that TDM-PM has the ability to accurately predict *not-relevant, maybe-relevant, and relevant* air traffic 88.7%, 48.5%, and 46.7% of the time, respectively. The confusion matrix, Table 1, shows the system performance compared to the pilot SME correlations

The results show that at this stage of development, TDM-PM was not successful at meeting our requirements.

This work identified numerous challenges of training a ML algorithm for pilot expertise replication:

- The volume of data collected was substantial, but perhaps not sufficient. Pilot interviews suggested numerous factors that were not included in this initial ML characterization that they deemed critical or important to determine "relevance" of surrounding air traffic.
- 79% of collected data was "Not Relevant" traffic. An additional study is in progress to assess if the database is biasing the algorithm or if alternative algorithms or training methods can provide better results.
- The current design would theoretically declutter more traffic (i.e., "not relevant") than human pilots would. The design criteria for matching relevant vs. non-relevant data is unknown but testing is on-going to obtain data to help inform these criteria.

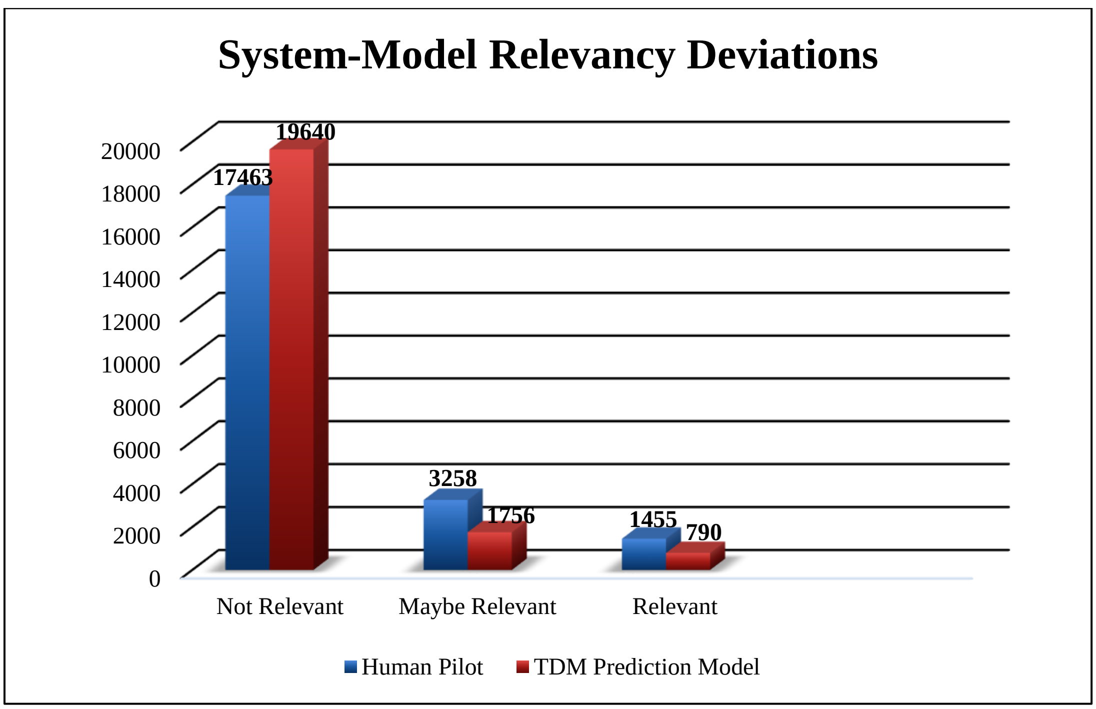


**Figure 3. Human-Pilot to TDM-PM Overall Relevancy Comparison**

## B. Model Prediction Latency Analysis

The rapidly changing traffic environments encountered in commercial flight means that the TDM-PM, with the purpose of autonomously decluttering the AC navigation display, must be able to make predictions quickly and communicate those predictions to the flight crew. However, there are limitations to the technology being developed and its use. Whether it is on the ground or airborne, autonomous systems struggle with information transmission issues, such as limited bandwidth, loss of communication, and particularly latency. Although the first two areas were compensated for in the study, latency remained an issue. With the size of the data set (≈22K data points) to process and learn from, the TDM-ML and TDM-PM were not able to overcome this issue, with a training time of 45sec and a prediction time of 24sec respectively. These results meant that the TDM-ML and TDM-PM were more than 8x our established latency threshold requirement. It should be noted, however, that both the learning algorithm and its prediction model were importing large amounts of data during a single iteration, along with the execution of a number lines of code, which attributed to the latency issues in both the learning and prediction processing of the model. Future research will address these issues.

**Table 1. Model to System One-to-One Response Comparison**

| | Pilot Relevancy Determinations | | | |
|---|---|---|---|---|
| **Model Predictions** | **Relevant** | **Maybe** | **Not Relevant** | |
| **Relevant** | **680**<br>3.06% | **73**<br>0.329% | **27**<br>0.122% | **86.1%** |
| **Maybe** | **168**<br>0.757% | **1581**<br>7.13% | **7**<br>0.032% | **90.0%** |
| **Not Relevant** | **607**<br>2.74% | **1604**<br>7.23% | **17429**<br>78.8% | **88.7%** |
| | **46.7%** | **48.5%** | **88.7%** | **61.3%** |

## C. TDM integrated with Sound Source Localization (SSL) & Navigation Display (ND)

Secondary to this technical evaluation was the integration of TDM-PM with a SSL algorithm and the cockpit ND, and to assess whether the addition of these sensory queues would improve a pilot's situation awareness. The SSL algorithm was developed in the Visual Studio C++ .NET framework and interacts with the Visual Imaging Simulator for Transport Aircraft Systems (VISTAS) simulator's 7.1 surround sound system, in order to localize auditory communication from ATC, relevant AC, and ownship. The TDM-PM transferred the state data of relevant AC to the SSL algorithm. This information helped map those AC to their respective aural channel, thereby, simulating intelligent party-line capabilities. The relevant state data was then highlighted to the flight crew by employing Light Craft (LC) software to render those AC onto the flight simulator's ND.

A more extensive test of the auditory integration of TDM-EFB with the 3D-Audio (SSL) technology will be conducted during a later, more in-depth study. In the meantime, during this study, the SMEs were surveyed as to whether TDM-EFB integration with SSL is more beneficial/desired as opposed to its integration with standard stereo quality amplification. 3) During the evaluation of the TDM-EFB, the ocular integration was limited, because highlighting of relevant AC was not possible.

## D. Pilot Cueing Enhancement (Auditory & Visual)

With respect to the additional sensory cues, evaluations showed that the pilots were very receptive to the integration of the ND of relevant traffic and speech-to-text data link functionality. The integration of SSL and TDM had mostly positive reviews from the pilots (97% approval) as they considered intelligent party-line capabilities to be a very good addition to the cockpit environment.

## IV. Conclusion

The purpose of this study was to carry out an initial assessment of the overall performance of the newly developed autonomous technology, comparing it to the human pilot expectations, and determining if certain factors could be achieved that contribute to establishing trust in the autonomous system. The results are used to determine what may be required in order to condition the algorithm and, subsequently, the prediction model for use in real world simulation and future implementation. The refinements to TDM-PM, from its prototype stage, are intended to help decrease pilot workload and improve a flight crew's overall situation awareness.

The one-to-one comparison of TDM-PM against the human-pilot system revealed that there is definite promise for the future development of the TDM-ML and TDM-PM in establishing much better accuracy and trust in in the PM. To accomplish this, first and foremost, the training of the TDM-ML should be equally applied to each category of relevance. This requires that the data used to classify an aircraft's relevance be input into the ML algorithm in equal amounts. Data collected to train the TDM-ML, during this study, were gathered from SMEs at a 3-to-1 ratio of *not-relevant* air traffic to *maybe-relevant* and *relevant* air traffic. This was key to the significant, predictions produced by TDM-PM with respect to *not-relevant* air traffic. The TDM-ML algorithm had three times the number of data points available for learning to distinguish what was *not-relevant* traffic as it did for learning what was *maybe-relevant* or *relevant*, i.e., it was bettered trained. This observation proved encouraging, however, with respect to TDM-PM achieving similar prediction accuracy in the case of *maybe-relevant and relevant* air traffic. This encouragement comes from the determination that instituting some additional steps and information to the training portion of the TDM-ML algorithm would provide the model with a better overall learning process. The future additions to the TDM-ML training process are as follows. First, the intent is to use an equal number of data points, representing the three different classes of relevance (*not-relevant*, *maybe-relevant, and relevant*) for training the TDM-ML. Second, additional data labels (predictors) are being explored as input parameters to the TDM-ML, such as altitude difference, vertical speed difference, and possibly AC closing rate. These parameters were identified by the SMEs as additional factors that pilots use to assess the relevancy of surrounding air traffic. Finally, a preprocessing step using an unsupervised learning (Clustering) algorithm will be implemented to reduce the prediction model's dimensionality, after which, the supervised learning (Classification) TreeBagger algorithm will use those results to obtain a more accurate predictive model. With the implementation of these steps, it is expected that the second phase of TDM development will result in the application yielding more accurate predictions and helping to define the necessary factors that lend themselves to establishing trust between humans and the model.

Concerning TDM latency issues, it is expected that latency will be significantly minimized during the second phase of application development. This improvement to TDM latency can be realized by implementing real-time streaming of the algorithm's input data, instead of conducting bulk data file reads. In addition, parallel computing will be a part of the learning algorithm's base structure. This will allow TDM-ML to process large amounts of data using a set number or all of the system server's core processors, thereby reducing, even further, training and prediction intervals as well reducing the time it takes to declutter the ND of irrelevant traffic. Implementing these adjustments, the expectation is that the TDM application will be able to achieve the minimum requirements that help instill trust in the model with respect to *model prediction latency*.

Future technical evaluation of TDM, as a fully integrated system, is to include both SSL and stereo transmission of intelligent party-line information, based on pilot preference. And, on a larger scale, the integrated system will include multiple learning algorithms, so as not to limit TDM-ML learning process.


## Acknowledgements

Author's thanks go to Randy Bailey (Flight Deck Interface Technology Group Team Lead), Jason R. Neuhaus (Senior Aerospace Engineer at NASA Langley Research Center), Trey Arthur (Senior Research Engineer at NASA Langley Research Center), Steve Williams (Senior Research Engineer at NASA Langley Research Center), and the entire Flight Deck Interface Team (FDIT).

## References


[1] NASA Aeronautics Airspace Operations and Safety Program Discussion with the ARTR http://sites.nationalacademies.org/cs/groups/depssite/documents/webpage/deps_152342.pdf

[2] MATHWORKS Machine Learning Algorithms https://www.mathworks.com/help/stats/supervised-learning-machine-learning-workflow-and-algorithms.html#bswlxh8

[3] Le Vie, L, R., Houston, V, E., "Development of Increasingly Autonomous Traffic DATA Manager Using Pilot Relevancy and Ranking Data", May 2017.

[4] MATHWORKS TreeBagger Class https://www.mathworks.com/help/stats/treebagger-class.html

[5] Dietterich, T, G., "Ensemble Methods in Machine Learning", 2001

[6] Breiman, L., "Bagging Predictors", Technical Report No. 421, September 1994

[7] Ribeiro, M, T., Singh, S., Guestrin, C., "Why Should I Trust You? Explaining the Predictions of Any Classifier", University of Washington, arXiv: 1602.04938v3, August 9, 2016

[8] Leetaru, K., "In Machines We Trust: Algorithms Are Getting Too Complex To Understand", Forbes Magazine, January 4,